Wireless Message Protocol 2.5.0 Specification

Abstract

This document defines version 2.5.0 of the Wireless Message Protocol, a specification designed to maximize efficiency and interoperability between wireless messaging networks. Based upon XML (Extensible Markup Language), an external short messaging entity can submit and query wireless messages.

Simplewire, Inc.
http://www.simplewire.com/

All Rights Reserved

This is an unpublished work. No part of this document may be copied, photocopied, reproduced, translated or reduced to any electronic or machine readable form without the prior permission of Simplewire, Inc.

Whilst the greatest care has been taken to ensure the accuracy of the information contained herein, Simplewire, Inc. does not warrant the accuracy of same. Simplewire, Inc. expressly disclaims all and any liability to any person, whether a purchaser of the publication or not, in respect of anything and of the consequences of anything, done or omitted to be done by any such person on reliance, whether whole or partial, upon the whole or any part of the contents of this publication.

This specification is a work in progress for the purposes of creating an open, non-proprietary, messaging standard between wireline and wireless networks. Simplewire, Inc. reserves the right to revise this publication and to make changes to it from time to time without obligation to notify any person or organization of such revision or changes.

Revision History

Version 2.5.0 – May 26, 2002

- Specified many additional client error codes for Internet connection errors.
- Added smart message submit message structure definitions.
- Added international addressing format.
- Added IRA encoding format.

Version 2.2.0 – March 27, 2001

- This is major update to and rewrite of the specification from the previous release. Changes span the entire document and are too numerous to list individually.

Version 1.0.0 – November 20, 2000

- Introductory specification released.

Table Of Contents

1 Introduction

1.1 Scope

This document is intended for designers and implementers of the Wireless Message Protocol. This document is useful for server and client developers creating or using applications that provide a Wireless Message Protocol interface.

1.2 Wireless Message

Wireless Messaging is the “pushing” of information to and from mobile devices, allowing users to receive and send time-sensitive notifications and maintain important contacts while on the move. It includes SMS (Short Message Service), EMS (Enhanced Message Service), and MMS (Multimedia Message Service). SMS involves sending a text-message of 140-160 characters to a cell phone or pager, and holds significant advantages over the use of mobile email for sending important information. Originally devised as an afterthought to notify users of incoming voice mail, SMS has grown to include the sending of ringtones and graphics, as well, and taken on a life of its own. In May of 2001 alone, the GSM Association reports that 19 billion SMS were sent throughout the world. Meanwhile, Cahners In-Stat Group predicts that this number will mushroom to 244 billion messages per month in 2004. EMS and MMS are technologies which will enhance wireless messaging with new capabilities, and the Wireless Message Protocol is designed to remain flexible with these emerging standards so that your applications stay on the cutting edge of the wireless industry.

1.3 Purpose

In today’s marketplace, it is impossible to build a wireless messaging initiative without interfacing with a server that consolidates the numerous fragmented messaging standards and carrier networks across the globe. The Wireless Message Protocol exists for this reason.

Getting a message off of the Internet and onto any carrier network is a significant endeavor, requiring hard-wired proprietary connections, infrastructure to speak with alternative messaging protocols and a protocol that provides a seamless integration into an application. The Wireless Message Protocol elucidates this complicated process through an XML (Extensible Markup Language) interface.

The Wireless Message Protocol supports communications over TCP/IP to one and two-way wireless devices. The protocol supports both the sending and receiving of textual messages as well as the sending and receiving of binary data such as ringtones, logos, formatted text, and much more. The protocol allows applications on dedicated or transient computers to send messages to wireless devices and determine delivery status.

1.4 Glossary

API	Application Programming Interface
ESME	External Short Message Entity.*
HTTP	HyperText Transfer Protocol
MS	Mobile Station
SME	Short Message Entity
SMSC	Short Message Service Centre
URL	Uniform Resource Locator
WMP	Wireless Message Protocol
XML	Extensible Markup Language

***** ESME exclusively refers to external sources of short messages such as message handling computers. It specifically excludes SMEs which are located within the mobile network such as a mobile station (MS).

2 Protocol Overview

WMP is based on the exchange of request and response XML between the ESME and the SMSC over an underlying TCP/IP network connection. The ESME may originate requests to the SMSC and receive responses back for those requests. All requests sent to the SMSC are immediately checked for preliminary errors and a valid response is returned.

2.1 Stateless Sessions

This version of WMP will only permit stateless sessions to be established between the ESME and the SMSC. Relevant subscriber information will be mandatory in each request sent to the SMSC so that each ESME request can be authenticated.

2.2 Network Layer

The underlying transport interface between the ESME and SMSC is based on a TCP/IP network connection. WMP is an application layer protocol and is not intended to offer transport functionality. Additional underlying transport layer protocols such as HTTP may be used to transmit WMP from ESME to SMSC.

2.3 HyperText Transfer Protocol Layer

HTTP can be used to transmit and receive WMP over TCP/IP networks between the ESME and SMSC.

2.3.1 Requests

WMP requests will be sent as an HTTP POST request to the SMSC. The "xml" parameter contained in the POST data shall be set to a properly url-encoded string of the WMP.

```
POST /wmp HTTP/1.1
Host: wmp-test.simplewire.com
Content-Length: 561

xml=%3C%3Fxml%20version%3D%221.0%22%20%3F%3E%0A%3Cr
equest%20version%3D%222.0%22%20protocol%3D%22paging
%22%20type%3D%22sendpage%22%3E%0A%20%20%20%20%3Cuse
r%20agent%3D%22Perl%2F2.00%22%20ip%3D%2264.94.350.8
0%22%2F%3E%0A%20%20%20%20%3Csubscriber%20id%3D%22JK
45J%22%20password%3D%22HEL343JLLJL%22%2F%3E%0A%20%2
0%20%20%3Coption%20method%3D%22asynch%22%20timeout%
3D%2230%22%20delimiter2%3D%22%22%2F%3E%0A%09%3Cpage
%20serviceid%3D%222%22%20pin%3D%221234567890%22%20f
rom%3D%22Joe%22%20callback%3D%221234567890%22%20tex
t%3D%22Hello%20World%22%2F%3E%0A%3C%2Frequest%3E%0A
```

2.3.2 Responses

WMP responses from the SMSC will be contained in the body of the HTTP response.

2.3.3 Errors

Any errors triggered from the HTTP protocol level such as 404 or 500 errors should be treated as network layer transmission errors. All errors generated on the WMP level will be returned as WMP responses in the body of the HTTP response.

3 Message Structures

The general format of all protocol messages, structures, and operations exchanged between the ESME and SMSC utilizes XML.

3.1 Request Header

A WMP request header specifies the xml version, request version, and request type. The request header will determine what response will be sent back from the SMSC. All operations must be contained within a request element.

```
<?xml version="1.0" ?>
<request version="2.0" protocol="paging" type="operation name">
</request>
```

3.2 Response Header

A WMP response header specifies the xml version, response version, and response type. All responses to operations are contained within a response element.

```
<?xml version="1.0" ?>
<response version="2.0" protocol="paging" type="operation name">
</response>
```

3.3 Operations

3.3.1 Submit

This operation is used by an ESME to submit a short message to the SMSC for onward transmission to a specified short message entity (SME). The submit operation can contain a text message or an enhanced message.

There are numerous optional parameters in a submit request which will determine what type of text message (e.g., 7-bit, 8-bit, or 16-bit (Unicode)) or what type of enhanced message (e.g., ringtone, picture, logo, etc.) will be sent.

Please refer to Section 5 – Examples for information regarding what optional parameters are used with what type of text or enhanced content.

3.3.1.1 Submit Request Syntax

The format of a submit request is defined in the following table:

	Element	Attribute	Description
Header	Request	Version	Identifies the version of the request. Set to "2.0"
	Request	Protocol	Identifies the protocol contained within the request element. Set to "paging"
	Request	Type	Identifies the type of request, or the operation. Set to "sendpage"

Mandatory	Subscriber	Id	Identifies the ESME account requesting this operation.
	Subscriber	Password	The password used by the SMSC to authenticate the ESME request.
	Page	Pin	Identifies the destination address (e.g., the recipient) of the short message. Please refer to Section 4.1 - International Address Format for information about the proper format for this field.
Optional	Page	Text	Up to 140 octets (160 7-bit characters) of short message user data. This is typically 7-bit text, but may contain Unicode (16-bit) text as well. Proper HTML escaped entities may be used in place of characters. HTML entities are typically used to send 16-bit characters in a 7-bit data channel. Please refer to the *Option DataCoding* field for more information on sending anything other than 7-bit data.
	User	Agent	Contains information about the user agent originating the request. This is mostly for statistical purposes, but can be used for enforcing SMSC access to specific user agents. While this is an optional parameter, an ESME SHOULD include this field with requests. An example would be “ActiveX/SMS/2.4.2”
	User	Ip	Contains information about the IP address of the user that the ESME request is on behalf of. This is mostly for statistical purposes, but could be used to limit access to a subset of IP addresses for the ESME.
	Page	Service Id	Allows the ESME to direct the SMSC to route message through a specific network route. Routes are unique, numeric ids assigned to each carrier (or network) that is supported by the SMSC.

	Page	Callback	Identifies the origination address (e.g., the originator or sender) of the short message. Please note that the SMSC will attempt to use this field, but it is not guaranteed. Please refer to Section 4.1 - International Address Format for information about the proper format for this field.
	Page	From	Sets a name for the sender (or originator) of the message. This field is PREPENDED to most messages and will be delimited from the Text field by the default “: “ (a colon followed by a space). Total message length adds the length of both the From and Text fields. The default delimiter can be overridden by setting a value in the *Opt Delimit*er field.
	Page	Ringtone	Used in place of or in addition to text message, when proper Option Type is set. Accepts RTTTL (Ringtone Text Transfer Language).
	Page	Image	Used in place of OR in addition to text message, when proper Option Type is set. Accepts IRA encoded hex dump of an image file. Currently accepted image formats include .BMP (Windows 95 Version), FlashPix, .GIF, .JPEG, .PNG, .PNM (ASCII or Raw Binary Data), and .TIFF. Please refer to Section 4.2 – IRA Encoding for information about the proper format for this field.
	Option	Country Code	GSM Mobile Country Code. Required for Operator Logos to be saved on phone. Please contact Simplewire Support for an up-to-date list of mobile country codes.

	Option	Data Coding	The option that tells the SMSC how to encode the text being sent to the mobile device. The values accepted are “7BIT”, “8BIT”, and “UCS2”. Encoding using 7 bits can represent Unicode characters in the range 0x0000 to 0x007F. Encoding using 8 bits can represent Unicode characters in the range 0x0000 to 0x00FF. Encoding using UCS2 can represent Unicode characters in the range 0x0000 to 0xFFFF. If the wrong setting is used (e.g. “8BIT” when “7BIT” is required), the text may not display correctly on the recipient’s mobile device.
	Option	Delimiter	Delimiter to use to separate the From and Text (and sometimes Callback) fields in the short message. Defaults to “: “ (a colon followed by a single space).
	Option	Flash	Either a “true” or “false” setting which requests the SMSC to flash or popup the message on the phone. In GSM networks, this actually means the message is not stored on the SIM and instead is immediately displayed on the user’s phone.
	Option	Network Code	GSM Mobile Network Code. Required for Operator Logos to be saved on phone. Please contact Simplewire Support for an up-to-date list of mobile network codes.
	Option	Phone	Identifies the phone manufacturer of the recipient’s MS. This field is currently used to format Enhanced (or Smart) messages. Currently, “nokia” is the only accepted format. Please note that while this is an optional parameter, it is REQUIRED for any type of enhanced content.
	Option	Type	Identifies what type of enhanced content is contained within the submit request. Failure to include this field simply defaults the SMSC to a text message. Possible settings include “ringtone”, “profile”, “icon”, “logo”, and “picturemessage”.

Please refer to Section 5 – Examples for correctly formatted WMP that shows a submit request for many types of text and enhanced content.

3.3.1.2 Submit Response Syntax

The format of a submit response is defined in the following table:

	Element	Attribute	Description
Header	Response	Version	Identifies the version of the response. Will be set to "2.0"
	Response	Protocol	Identifies the protocol contained within the response element. Will be set to "paging"
	Response	Type	Identifies the type of response, or the operation response. Will be set to "checkstatus"
Mandatory	Error	Code	Indicates the success or failure of the submit request. The complete list of error codes is available at http://www.simplewire.com/developers/knowledge/reference/errors/
	Error	Description	A textual description of the numeric *Error Code* field which can be used to quickly debug its meaning.
	Error	Resolution	A textual description of ways that the Error Code can be overcome.
Optional	Ticket	Id	
	Ticket	Transmit	Number of SMS required to fully transmit entire submit request to MS. For example, some Smart Messages may need to be split up into 1-3 separate SMS "chunks" to be fully sent to the destination MS.
	Ticket	Price	Number of credits required to send each SMS.
	Ticket	Total	Total number of credits deducted from account balance.
	Ticket	Pin	Destination address of MS that will receive the message. Typically, this should be used to confirm the submit request was sent to the correct MS.

Please refer to Section 5 – Examples for correctly formatted WMP submit responses.

3.3.2 Query

This operation is issued by the ESME to query the status of a previously submitted short message to the SMSC. The mechanism used for matching is based on a unique ticket id returned in a submit response.

3.3.2.1 Query Request Syntax

The format of a query request is defined in the following table:

	Element	Attribute	Description
Header	Request	Version	Identifies the version of the request. Set to "2.0"
	Request	Protocol	Identifies the protocol contained within the request element. Set to "paging"
	Request	Type	Identifies the type of request, or the operation. Set to "checkstatus"
Mandatory	Subscriber	Id	Identifies the ESME account requesting this operation.
	Subscriber	Password	The password used by the SMSC to authenticate the ESME request.
	Ticket	Id	*Ticket Id* of the short message whose status is to be queried. This must be the SMSC assigned *Ticket Id* allocated to the original short message when submitted to the SMSC by the submit operation and returned in the submit response.
Optional	User	Agent	The *User Agent* field contains information about the user agent originating the request. This is mostly for statistical purposes, but can be used for enforcing SMSC access to specific user agents. While this is an optional parameter, an ESME SHOULD include this field with requests. An example would be "ActiveX/SMS/2.4.2"
	User	Ip	The User Ip field contains information about the IP address of the user that the ESME request is on behalf of. This is mostly for statistical purposes, but could be used to limit access to a subset of IP addresses for the ESME.

Please refer to Section 5 – Examples for correctly formatted WMP showing a query request.

3.3.2.2 Query Response Syntax

The format of a query response is defined in the following table:

	Element	Attribute	Description
ade	Response	Version	Identifies the version of the response. Will be set to "2.0"

	Response	Protocol	Identifies the protocol contained within the response element. Will be set to "paging"
	Response	Type	Identifies the type of response, or the operation response. Will be set to "checkstatus"
Mandatory	Error	Code	Indicates the success or failure of the query request. If the request was successful then the Status Code, Status Description, and Status Resolution fields will be present in the response. The complete list of error codes is available at http://www.simplewire.com/developers/knowledge/reference/errors/
	Error	Description	A textual description of the numeric *Error Code* field which can be used to quickly debug its meaning.
	Error	Resolution	A textual description of ways that the Error Code can be overcome.
Optional	Status	Code	Indicates the current state of the queried message. The state can either be temporary or final depending on the actual code. The complete list of status codes is available at http://www.simplewire.com/developers/knowledge/reference/errors/
	Status	Description	A textual description of the numeric *Status Code* field which can be used to quickly debug its meaning.
	Status	Resolution	A textual description of ways that the *Status Code* can be overcome.

Please refer to Section 5 – Examples for correctly formatted WMP that shows a query response.

4 System Definitions

4.1 International Address Format

WMP defines an international format for addressing a mobile station or SME. A well-formed international address includes a format specifier, followed by the country code and national trunk.

A format specifier shall be a ‘+’ (plus) character.

A country code is the numeric 1, 2, or 3 digit code assigned to each country throughout the world. For example, the United States and Canada use the country code of ‘1’, whereas the United Kingdom uses the country code of ‘44’.

A national trunk is the rest of the address that is used by the country to determine where to route the message. This can be of variable length based on the National Numbering Plan used within the country code. Please note that ALL LEADING ZEROES should be removed from the national trunk.

The following are acceptable international address formats:

```
+11005101234  ->     (‘+’, ‘1’, ‘1005101234’)
+13135551212  ->     (‘+’, ‘1’, ‘3135551212’)
+44123456789  ->     (‘+’, ‘44’, ‘123456789’)
+6598482412   ->     (‘+’, ‘65’, ‘98482412’)
```

4.2 IRA Encoding

Binary (8-bit) data sent over 7-bit communication channels requires the conversion of binary data into IRA encoded characters. Each 8-bit octet is represented by two IRA (7-bit) encoded characters. Please refer to the following table:

Binary Data	Hex Value	IRA Characters
10101010	AA	A and A
11111111	FF	F and F

Thus, each byte is represented by two 7-bit characters. This payload reduction is necessary, as in 8-bit sending the largest representable hex value is 0xFF, but in 7-bit sending each character can only represent hex values up to 0x7F.

5 Examples

5.1 Submit Operation

5.1.1 Text Message

In this example, a 7-bit text message (from=”Demo” and text=“Hello World!”) will be sent using the Option Flash field to instruct the SMSC to immediately display the message on the MS. Please note that complete HTML entity escaping was used even though all characters can be represented in 7-bit data. Please refer to the Unicode Text Message example to see how to HTML entity escape only certain characters.

5.1.1.1 Submit Request

```
<?xml version="1.0" ?>
<request version="2.0" protocol="paging" type="sendpage">
        <user agent="ActiveX/SMS/2.4.2" ip=""/>
        <subscriber id="100-100-100-10000" password="A23F67"/>
        <option flash="true"/>
        <page pin="+11005101234"
              from="&#68;&#101;&#109;&#111;" callback="+13136214123" text="
        &#72;&#101;&#108;&#108;&#111;&#32;&#87;&#111;&#114;&#108;&#100;&#
        33;"/>
</request>
```

5.1.1.2 Submit Response

```
<?xml version="1.0"?>
<response version="2.0" protocol="paging" type="sendpage">
        <error code="2" description="Message successfully queued." resolution=""/>
        <ticket id="LTIQH-0FQY5-PJ363-IEC4I" transmit="1" price="1.0" total="1.0"
        pin="+11005101234"/>
</response>
```

5.1.2 Unicode Text Message

In this example, a 16-bit text message (text=”Hello World – Ā”) will be sent using the Option Data Coding field. This example also demonstrates example usage of the Page Service Id field which instructs the SMSC to use a specific network route. Please note that the “Ā”

5.1.2.1 Submit Request

```
<?xml version="1.0" ?>
<request version="2.0" protocol="paging" type="sendpage">
        <user agent="ActiveX/SMS/2.4.2" ip=""/>
        <subscriber id="100-100-100-10000" password="A23F67"/>
        <option datacoding="UCS2"/>
        <page pin="+11005101234"
              from="Demo" callback="+13136214123" text="Hello World! - &#256;"/>
</request>
```

5.1.2.2 Submit Response

See 5.1.1.2

5.1.3 Ringtone Smart Message

In this example, a ringtone will be sent to a Nokia phone.

5.1.3.1 Submit Request

```
<?xml version="1.0" ?>
<request version="2.0" protocol="paging" type="sendpage">
		<user agent="ActiveX/SMS/2.4.2" ip=""/>
		<subscriber id="100-100-100-10000" password="A23F67"/>
		<option type="ringtone" phone="nokia"/>
		<page pin="+11005101234"
		ringtone="Simplewire:d=4,o=5,b=63:8a,8e,32a,32e,16a,8c6,8a,32c6,32a,16c6,8e6
,8c6,32e6,32c6,16e6,8g6,32g,32p,16g,32c6,32g,16c6,8e6,32p"/>
</request>
```

5.1.3.2 Submit Response

See 5.1.1.2

5.1.4 Icon Smart Message

In this example, a caller-line icon will be sent to a Nokia phone. The image used is the “example.gif” file available with all Simplewire SMS SDK downloads.

5.1.4.1 Submit Request

```
<?xml version="1.0" ?>
<request version="2.0" protocol="paging" type="sendpage">
		<user agent="ActiveX/SMS/2.4.2" ip=""/>
		<subscriber id="100-100-100-10000" password="A23F67"/>
		<option type="icon" phone="nokia"/>
		<page pin="+11005101234"
		image="47494638376148000E00800000000000FFFFFF21F90400000000002C00000
00048000E0040027C8C8FA97B000CA34CCED450F3C3B7EFAD3C22D65821E478667A711CD8
B2A5497F6E934E7AABF7BEC4FBDD84441CE5234A5A543710E909B5456954D9F2AAC1C6603
904CC4BF576514E61B0884E1FD5441672E946BAAFAFEEB7469786C7AAEC90195732725756B
765F7E617F766F8353853E854D836C966F97376295400003B"/>
</request>
```

5.1.4.2 Submit Response

See 5.1.1.2

5.1.5 Profile Smart Message

In this example, a downloadable profile will be sent to a Nokia phone along with the name of “Simplewire”. The image used as the screen saver is the “example.gif” file available with all Simplewire SMS SDK downloads.

5.1.5.1 Submit Request

```
<?xml version="1.0" ?>
<request version="2.0" protocol="paging" type="sendpage">
            <user agent="ActiveX/SMS/2.4.2" ip=""/>
            <subscriber id="100-100-100-10000" password="A23F67"/>
            <option type="profile" phone="nokia"/>
            <page pin="+11005101234" text=”Simplewire”
            ringtone="Simplewire:d=4,o=5,b=63:8a,8e,32a,32e,16a,8c6,8a,32c6,32a,16c6,8e6
,8c6,32e6,32c6,16e6,8g6,32g,32p,16g,32c6,32g,16c6,8e6,32p"
            image="47494638376148000E00800000000000FFFFFF21F90400000000002C00000
00048000E0040027C8C8FA97B000CA34CCED450F3C3B7EFAD3C22D65821E478667A711CD8
B2A5497F6E934E7AABF7BEC4FBDD84441CE5234A5A543710E909B5456954D9F2AAC1C6603
904CC4BF576514E61B0884E1FD5441672E946BAAFAFEEB7469786C7AAEC90195732725756B
765F7E617F766F8353853E854D836C966F97376295400003B"/>
</request>
```

5.1.5.2 Submit Response

See 5.1.1.2

5.1.6 Logo Smart Message

In this example, an operator logo will be sent to a Nokia phone. The image used is the “example.gif” file available with all Simplewire SMS SDK downloads.

5.1.6.1 Submit Request

```
<?xml version="1.0" ?>
<request version="2.0" protocol="paging" type="sendpage">
            <user agent="ActiveX/SMS/2.4.2" ip=""/>
            <subscriber id="100-100-100-10000" password="A23F67"/>
            <option type="logo" phone="nokia" countrycode="310" networkcode="16"/>
            <page pin="+11005101234"
            image="47494638376148000E00800000000000FFFFFF21F90400000000002C00000
00048000E0040027C8C8FA97B000CA34CCED450F3C3B7EFAD3C22D65821E478667A711CD8
B2A5497F6E934E7AABF7BEC4FBDD84441CE5234A5A543710E909B5456954D9F2AAC1C6603
904CC4BF576514E61B0884E1FD5441672E946BAAFAFEEB7469786C7AAEC90195732725756B
765F7E617F766F8353853E854D836C966F97376295400003B"/>
</request>
```

5.1.6.2 Submit Response

See 5.1.1.2

5.1.7 Picture Smart Message

In this example, a picture message will be sent to a Nokia phone along with the text “Hello”. The image used is the “example.gif” file available with all Simplewire SMS SDK downloads.

5.1.7.1 Submit Request

```
<?xml version="1.0" ?>
<request version="2.0" protocol="paging" type="sendpage">
            <user agent="ActiveX/SMS/2.4.2" ip=""/>
```

```
        <subscriber id="100-100-100-10000" password="A23F67"/>
        <option type="picturemessage" phone="nokia"/>
        <page pin="+11005101234" text="Hello"
        image="47494638376148000E00800000000000FFFFFF21F90400000000002C00000
00048000E0040027C8C8FA97B000CA34CCED450F3C3B7EFAD3C22D65821E478667A711CD8
B2A5497F6E934E7AABF7BEC4FBDD84441CE5234A5A543710E909B5456954D9F2AAC1C6603
904CC4BF576514E61B0884E1FD5441672E946BAAFAFEEB7469786C7AAEC90195732725756B
765F7E617F766F8353853E854D836C966F97376295400003B"/>
</request>
```

5.1.7.2 Submit Response

See 5.1.1.2

5.2 Query Operation

5.2.1 Query Message

In this example, a previously submitted message with a Ticket Id of “FIQ3K-WUSTK-P0UCI-IRH9E” is queried.

5.2.1.1 Query Request

```
<?xml version="1.0" ?>
<request version="2.0" protocol="paging" type="checkstatus">
        <user agent="ActiveX/SMS/2.4.2" ip=""/>
        <subscriber id="100-100-100-10000" password="A23F67"/>
        <ticket id="FIQ3K-WUSTK-P0UCI-IRH9E"/>
</request>
```

5.2.1.2 Query Response

```
<?xml version="1.0"?>
<response version="2.0" protocol="paging" type="checkstatus">
        <error code="0" description="Request ok." resolution=""/>
        <status code="4" description="Message successfully delivered." resolution =""/>
</response>
```